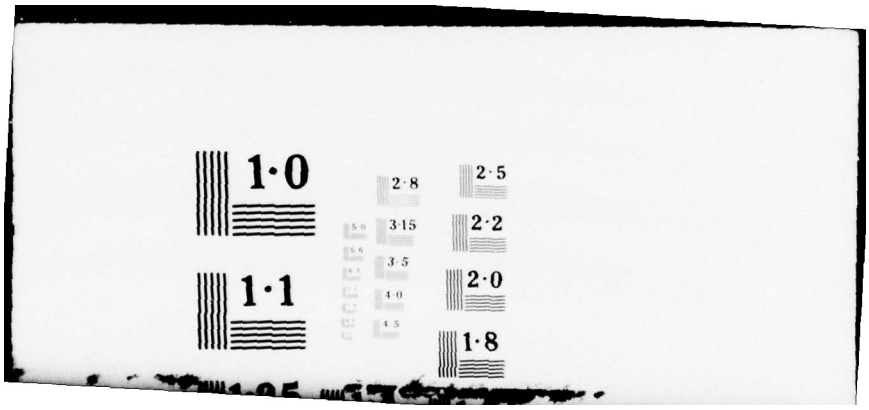
AD-A044 437 ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE F/G 6/4
REPRESENTING THE ORIENTATION OF DENDRITIC FIELDS WITH GEODESIC --ETC(U)
SEP 76 C M BROWN N00014-75-C-1091
UNCLASSIFIED TR-13 NL

1 OF 1
ADA
044437

END
DATE FILMED
10-77
DDC


1·0
1·1
2·8
3·15
3·5
4·0
4·5
5·0
5·6
6·3
7·1
8·0
9·0
2·5
2·2
2·0
1·8

電腦科學

Department of Computer Science
University of Rochester
Rochester, New York 14627

REPRESENTING THE ORIENTATION OF DENDRITIC FIELDS WITH GEODESIC TESSELATIONS

C.M. Brown
Computer Science Department
The University of Rochester

TR13

SEPT 1976

37 p.

The orientation properties of a population of three-dimensional vectors may be described by using "geodesic dome"-like constructions as the basis for histograms of the data. This work is complementary to past efforts to describe the orientation of structures of (undirected) rods and presages future work in statistical tests on vector data. An application to neuroanatomy is discussed, and two computer systems for acquisition/analysis and analysis/display of such data are described.

The preparation of this paper was supported in part by the Advanced Research Projects Agency of the Department of Defense, and was monitored by ONR under Contract No. N00014-75-C-1091 and in part by the Alfred P. Sloan Foundation under Grant No. 74-12-5.

410 326

Contents

1. The Problems.

> One of the difficulties in understanding the function of the brain is that it is like nothing so much as a lump of porridge.
>
> --R.L. Gregory

1.1. Dendritic Orientation.

In this paper we explore the concept of the "orientation" of the dendritic field of a neuron. In this Section we will say why the concept of dendritic orientation may be important, how computers are being used to aid in quantititive work in neuroanatomy, and what aspects of orientation we are trying to get at with our measures.

The thought that some concept such as "dendritic orientation" may be important arises from:

1) the theory that the arrangement and extent of a neuron's synapses are correlated with its trigger features.

2) the fact that a topological ordering is preserved from the retina to the lateral geniculate nucleus to the visual cortex. Thus LGN terminals projecting from retinal cells adjacent along some axis would contain synapses with the dendritic field of a visual neuron that lie on an axis identifiable with the axis on the retina.

3) the evidence that there is neural plasticity in some developing (or even adult [Creutzfeldt and Heggelund]) organisms. The functional properties of individual neurons, such as the orientation of their visual field [Hubel and Wiesel], can be modified by controlling for a time the classes of visual stimuli which are available to the organism.

Taken together, the theories, facts, and evidence motivate the conjecture that not only can the functional characteristics of neurons be changed by regimes of visual deprivation, but their gross anatomical, geometric characteristics may be changed as well. For instance, if kittens during their early development are only allowed to see visual patterns consisting of bars of light parallel to a single direction relative to their retinae, we may find evidence thereafter that the dendrites of their visual neurons have in some geometric sense actually taken on an anatomical, physical orientation which can be correlated with the orientation of the bars of light. It is these conjectural, unspecified geometric properties of the dendritic field of the neuron that we call "dendritic orientation." This paper is concerned with two measures of quantities which may reflect dendritic orientation. Most of its technical content refers to the concept of "microstructure"; the reader interested in the

technical details of the "macrostructure" measure is referred to [Brown].

1.2. Computer Analysis of Neuronal Structure.

The digital computer has become an increasingly important tool for investigation of the structure of neuronal processes. Several laboratories have automated microscopes, and have used the computer to reconstruct and to some degree analyse the 3-D structure of neurons [Wann et al., Reddy et al., Garvey et al.]. Most of the laboratories are also interested in the real-time use of the computer to aid in the data-acquisition process. In this application the machine takes away from the operator some part of the task of deriving metric 3-D information from a microscope slide (or slides). Usually the machine keeps track of the coordinates of the point in the slide under examination, and upon commands from the operator can remember the point, its topological relation to other points, etc. Sometimes the machine is more autonomous, actually tracking the neuronal process through a thick section of tissue and remembering the points it has seen. It is a system such as this which is briefly described in Section 3.1. More information is available in [Garvey et al.]. The dendrite tracking program is not fully autonomous; when in doubt about how to proceed (for instance, if it is tracking a dendrite and another crosses at a small angle) it will ask the operator for help. The 3-D data and topological connectivity data are stored on disk for later analysis.

The information stored by all the existing programs is basically equivalent; it allows the structure of the complex system of branches of the neural dendrites to be reconstructed in 3-D. In the system described here, the dendrites are stored as "rod trees" (Figure 1.1) in which the primitive elements are rods, one-dimensional segments specified by their endpoints. (Thus the system stores no data on the thickness of a dendrite). These rods placed end-to-end in space form a 3-D tree structure in the shape of the neural processes. Each rod results from a tracking movement from one point to another by the dendrite tracker. In the sequel, "rod" will refer to one of these primitive tree elements, not to a piece of retinal anatomy.

INSERT FIGURE 1.1. HERE

If the computer analysis of this information does not proceed beyond this point, and the computer is merely used as a display device to show the neuron in stereo, or perhaps dynamically rotating with selected dendrites blinking or separated, the computer provides a valuable tool for research. Many projects content themselves with this use of the computer for analysis, perhaps also computing a few properties such as the length of dendrites, etc.

Obviously, however, the geometrically accurate description within the machine allows metric, quantitative parameters of the neuron to be extracted. It is easy to conjure up several simple

geometric/topological properties as some measure of the volume and spatial extent of the dendritic field, the branching properties of dendrites in it, etc. Further, over a neuron or over a population of them, statistics can be gathered to relate these parameters to one another. The main stumbling block here is that it is not particularly easy to think of meaningful questions to ask about the geometrical structure of neurons aside from the ones answered by the accurate representation of the neuron in 3-D. We do not understand what physical parameters to measure, in general, and because of this the computer has not been exploited to a very great degree to analyze the data it has helped to gather about neurons.

This paper is the result of a computer scientist cooperating with a neuroanatomist to produce compact characterizations of some physical properties of the complex neuronal dendritic structure. The idea is to produce a succinct description of certain neuronal properties so that the complex physical reality is reduced to a few numbers which can easily be compared. The measures mentioned in this paper are meant to characterize "dendritic orientation." They are an attempt to impose a model on the data and to extract parameters from it that are quantified and of recognized physical significance. If the parameters turn out to be inadequate to the task, and indeed do not reflect the important physical facts about dendritic orientation, that will not be too surprising. They may find use elsewhere (since we can say exactly what they mean to characterize), and in any event they represent a step toward more advanced computer analysis of neuronal properties which may lead others to take more steps.

1.3. Macrostructure and Microstructure.

To explain the two aspects of 3-D orientation we have chosen for study, we refer to the four 2-D structures shown in Figure 1.2.

INSERT FIGURE 1.2. HERE

Since we do not know just how dendrites may be affected by visual deprivation, we cannot falsify the vague conjecture made in Section 1.1. But if a measure of orientation when applied to the neurons revealed the correlations we are looking for, the conjecture would to some degree be verified. Starting in the simplest possible way to look for orientational parameters, we came up with the macrostructure measure. Since it measures only the most gross properties of the dendritic field, and is inadequate to differentiate cases in which we would all say that the "orientation" of the structure was different, some other measure seemed necessary. The two measures we now have, taken together, can distinguish the sorts of orientation properties manifested by the diagrams of Figure 1.2; it remains to be seen whether they are adequate to the neurological task.

We desired that the macrostructure be characterized by a terse description of global properties of the system of dendrites. The microstructural description should be based on properties of smaller

parts of the structure. To obtain a compression of information, if not a description as terse as that of the macrostructure, it seems attractive to use some statistical way of describing how the ensemble of smaller parts behaves. Not surprisingly for this paper, the global properties of the dendritic field we use have to do with elongation or dispersion in space, and the local properties of the smaller parts of the structure we use describe orientation of the smaller part (in fact, of rods).

2. The Solutions.

> Great circles are the shortest distance around spheres and they are intertriangulated so they're in the most comfortable position they could possibly get into.
>
> --Buckminster Fuller

2.1. Macrostructure and Principal Components.

If the dendrites of the neuron are considered merely as points (or rods) in space, they form a swarm whose general shape may be dispersed in one direction more than another. One of the simplest measures for this is the moment of inertia, which is actually quite closely related [Pearson] to a more evocative measure, which can be described as moments about planes normal to the principal axes of inertia [Christie]. This measure provides us with the direction in 3-D along which the swarm is "most dispersed" (according to a mass*distance**2 measure of the contribution a point makes to the dispersion from a plane). It also gives numbers expressing the relative degree of dispersion about the two planes normal to the principal one. By this measure, structures like A and B of Figure 1.1. are indistinguishable, but A, C, and D all have different descriptions and so are distinguishable. The macrostructure description we use [Brown] consists of three perpendicular vectors describing dispersion in three principal directions. The moments are computed both around the soma of the neuron and about its center of mass.

This measure has certain disadvantages inhering in its elegance (or taciturity...only seven independent numbers result from the entire macrostructure analysis) and in the questionable relationship of the numbers to spatial extent (since they express moments, which depend on distances squared instead of just distance). The principal components do seem to reflect accurately some of our qualitative feeling about the direction of elongation of a point swarm (or rod swarm), are needed to differentiate between structures like C and D of Figure 1.1., and thus form an important part of the data analysis.

Figure 2.1. shows some results of the principal component analyser.

INSERT FIGURE 2.1. HERE

2.2. Microstructure and a 3-D Histogram of Directions.

The aspect of orientation with which this paper is mostly concerned is the microstructural one. We take this to be a description of the ensemble behavior of small primitive parts of the dendrite. We choose the rods from which the rod tree (Figure 1.1.) is made to be our primitive elements. Each of these is considered to be

directed from the soma out towards the periphery of the dendrite. We disregard its position in 3-space, and only attribute to it the properties of 3-D direction and length, to get a description consisting of 3 independent numbers. Attribution of a directed orientation to the elements is the general case; it easily specializes if one wants the elements to "point both ways" (i.e., neither way). Undirected orientational data is called "axial data," and can be considered as distributed over a hemisphere instead of a sphere. Everything in this paper is easily adaptable to such data.

Leaving aside for now the state of the art on the statistics of directional data (but see Section 2.3), we look for a way to condense the raw data of our experiment into a more tractable form which does not throw away as much information as the macrostructural method. One answer [Mardia] is to use principal components once more, only this time to describe a set of weighted points on a sphere instead of an extended rod structure. The points are on the sphere in the direction (from the origin) of the vectors, and weighted by vector length. This sort of description is used to get at gross characteristics of the data, such as whether it exhibits a girdle or unimodal distribution, but as we have seen it does not provide a detailed description of shape.

The approach we choose instead is basically to cluster the vectors and report about the clusters. That sort of strategy will clearly differentiate A and B in Figure 1.1., because the number of clusters will be different. B, C, and D could be cleverly constructed so as to appear identical in this microstructural aspect, of course.

The sort of clustering we chose was simply to histogram the directions in 3-D and present the histogram as the microstructural description. That is, we partition 3-D directions into a number of "bins," associate a magnitude with each bin, and add the length of each directed rod into the magnitude of the bin which contains that rod's direction. After all rods have been dealt with, we have a number of ordered pairs of the form (bin, magnitude), or (range of directions, total amount of dendrite in direction range), and we take this to be our description of the dendrite. The total amount of information possible in the description depends on how finely we divide the set of directions; the system we describe can divide 3-D directions into partitions of cardinality 10*n**2 + 2 for n between 1 and 8. If every histogram bin has a nonzero magnitude, then the maximum length of description would be 3*(10*n**2 +2) numbers (leaving aside the fact that the histogram bin identifiers, which are vectors, are not independent).

This is all very well: how do we construct the histogram? A histogram is used to chart the relative frequencies of occurrence of some measurement. In its usual form as a bar graph, it approximates a probability distribution function (which it becomes when its area is kept to unity while the bin width becomes infinitesimal).

I have found no other attempt to construct a meaningful histogram of 3-D orientational data. One moderately useful idea is to use a projection (Lambert's equal area projection is good) to map the 3-D

data onto a disk in the plane. Such a display may be helpful, but it lacks visual appeal and is not useful for histogramming.

Two-dimensional directional data may be histogrammed on the real line, as shown in Figure 2.2.

INSERT FIGURE 2.2. HERE

Another display is the "circular histogram" (Figure 2.3.) in which the bar histogram is wrapped around a circle of some "bias diameter"; the bias radius represents zero frequency.

INSERT FIGURE 2.3. HERE

The "rose diagram" (Figure 2.4.) is another obvious representation.

INSERT FIGURE 2.4. HERE

If a point is associated with the middle of the arcs in a rose diagram or circular histogram and the points are connected, a form of histogram results which is parallel to that we will demonstrate for use with 3-D data (Figure 2.5.).

INSERT FIGURE 2.5. HERE

So far we have taken it for granted that histogram bins were of the same width (angular increment); any 3-D histogram should have the same features. Each bin should enclose equal solid angle, the bins should be congruent, and each bin center should be separated from the centers of its neighboring bins by a uniform amount.

When we try to achieve these ideals, we quickly rediscover what Euclid knew...there are but 5 ways to get congruent bins equiangularly spaced around the sphere! The 5 ways are of course the Platonic solids. This paper is concerned with constructions based on the icosahedron, but by the duality of the dodecahedron and icosahedron, some of the information in Section 4 can be taken as describing the dodecahedron as well (Figure 2.6.).

INSERT FIGURE 2.6. HERE

Our desire is to find sets of points on the surface of a sphere which are equally separated from one another. These points will serve as the centers of our histogram bins. We desire more points than the 12 provided by the vertices of the icosahedron, or the 20 provided by

its faces (i.e. the vertices of the dual dodecahedron). This desire forces us to approximate solutions; the strategy is to start with the icosahedron and to subdivide each of its equilateral triangle faces into approximations to equilateral triangles. These new triangles are "punched out" to a uniform distance from the center of the polyhedron, where they become facets of a faceted polyhedron which approximates a sphere as the fineness of subdivision increases (Figure 2.7.). As it happens, this construction was done by Buckminster Fuller in the development of his geodesic domes [Fuller]. If the triangles of the faceted polyhedron are projected onto the surface of the sphere, they become spherical triangles which are approximately congruent and which tesselate the surface of the sphere. It is this tesselation which has induced the desirable properties on the histogram bins, and to which we refer in the title.

INSERT FIGURE 2.7. HERE

The subdivision of the icosahedron provides faceted polyhedra of 20*(n**2) facets and 10*(n**2) + 2 vertices (n = 1,2,...). Rather than consider the facets as histogram bins, we interpret the vertices as centers of histogram bins. In the histogramming process, each data vector is associated with the vertex which is closest to the direction of the vector. The magnitude of the data vector is added to an accumulating magnitude associated with the vertex. When all the data vectors have been processed, each vertex is stretched away from the center by an amount proportional to its accumulated magnitude plus some bias radius, thus forming an approximate 3-D equivalent to the histogram form of Figure 2.4. Views of distorted faceted polyhedra are shown in Figure 2.8.

INSERT FIGURE 2.8. HERE

2.3. The Future.

Work on the statistics of directional data does exist [Mardia]. A later paper will look more carefully at the proper statistical methods for testing the anatomical conjecture of Section 1.1. What seems to be needed is a nonparametric two independent sample test on data from spherical distributions. I know of no such test at present; the difficulty arises from the lack of distribution-free statistics for multivariate data. We are devoting some effort to determining how to use existing tests (such as Chi-squared) on this data (cf. [Siegel]), and to developing new ones (around the permutation test). These general statistical developments are not meant to substitute for intelligent observation of the data, which may yield meaningful parameters whose influence may be verified by specific tests.

In this latter regard, the data saved by the dendrite tracker is adequate to answer many questions about the dendrites which are not at present addressed. In particular, information about the actual spatial extent of the fields, topological information about branching

ratios, information on the length of processes, etc., can all be extracted from the raw data.

The use of the computer allows preservation of large amounts of data (such as that needed to characterize a complex structure like neuronal dendrites) and analyses and interpretations of that data which are limited virtually only by the imagination of the investigator. It is not clear what parameters are important in the concept of neural orientation; whether or not the particular measures we have picked are germane and illuminating, the computer is such a powerful and versatile tool that is bound to be increasingly important in the quantitative description of complex biological structures and in the testing of models which attempt to isolate important parameters from such masses of data.

3. The Systems.

Ay, yi, yi! Is dis a system?

--Milt Gross

3.1. Acquisition of Data and Macrostructure Analysis.

Three-dimensional data describing the neural dendrites is gathered using an automated microscope commanded by an automatic dendrite-tracking program [Garvey et al.]. Alternative methods of reconstructing the neuron in 3-D involve piecing together images of serially sectioned slides [Wann et al., Reddy et al.].

The data-gathering system is based around a Nova computer with 32K words of core. The configuration of the system is shown in Figure 3.1.

INSERT FIGURE 3.1. HERE

One side of the binocular microscope gives input to the image dissector and the other is used to project an image of the field of view onto the oscilloscope screen, which can simultaneously display the actions of the dendrite tracker. The image dissector gives input to the dendrite tracking program, which writes to disk a record of where it has been in co-ordinates related to the control systems for the stage and focus adjustments. The tracker accepts data from the operator which serves to identify true (brain-coordinates) anterior-posterior and vertical directions. For analysis, the disk-stored data are converted into x,y,z coordinates (with units of microns) relative to the cell body of the neuron; they can be further converted into coordinates relative to the center of mass of the dendritic field. The system has the capability of displaying on an oscilloscope screen the dendritic field as it is stored on disk.

Using the converted coordinates, an ALGOL program determines the principal components of the dendritic field. Thus the analysis of the macrostructure of the dendritic field is completed at the site of data acquisition. The results of this analysis are numerical and quite compact, and are immediately available through the terminal or printer.

The same conversion routines are used to make a disk file of data in true x,y,z coordinates for use by the system in the Computer Science Department which analyses the microstructure of the dendritic field by histogramming its component directions. This disk is readable by the Eclipse computer at the Computing Laboratory, and may be sent along the Ethernet to the ALTO computer which performs the analysis and display.

3.2. Microstructure Analysis and Display of Results.

3.2.1. The Hardware System.

Faceted polyhedra are constructed and displayed using an ALTO minicomputer, which is connected in a high-speed network to other ALTOs, printers, etc. in the computing laboratory of the Computer Science Department (Figure 3.2).

INSERT FIGURE 3.2. HERE

The ALTO is a 64k word computer with its own 1.2 MWord disk and a 640 x 800 raster black and white display which is run by a display processor incorporated within the CPU microcode. One bit in the memory corresponds to one display element. The display is generally more flexible than is a vector display, since textures, shaded pictures, etc. can be put up on it. For this application, we only needed a line-drawing capability, and a locally written graphics package was used. All programming was in BCPL [Richards]; the floating point calculations are done in software.

3.2.2. The Software System.

The display system uses a data file produced by the data-acquisition system to make histograms which may then be viewed, transformed (cf. [Newman and Sproull] for a treatment of the techniques of geometric transformations for display), plotted on paper, etc., in response to the user's commands; thus it is interactive, but not on-line with data gathering.

The user arrives at the ALTO, inserts a disk pack which has the display system and a data file on it, and activates the system.

At the TOP of the system (cf. the t command below) he is asked how finely subdivided a faceted polyhedron he wants. He may choose from 1 to 7 divisions along the edge of the icosahedral faces, thus getting between 12 and 492 histogram bins.

He may then issue commands to the system to bring about various actions involving parameters of the system, display, histogramming, plotting, etc. All parameters are initialized to reasonable default values. The commands are:

any character but those below: display the current version of the faceted polyhedron. A set of axes is also displayed to keep track of image rotations.

? (HELP): type out the list of legal commands.

b (BIASDIAMETER): system asks for integer giving the

magnitude (screen coordinates) assigned to empty histogram bins. The smaller it is, the spinier the histogram looks.

s (SIZE): system asks for integer giving the maximum size (screen coordinates) of a histogram bin (maximum magnitude of a vertex).

e (EYESPACING): system asks for integer giving separation between left and right images in stereo pair displays.

f (FOCALLENGTH): system asks for integer giving a "focal length" which describes the extent of perspective distortion to apply to the image. Nonpositive focal lengths are interpreted to mean orthographic projection.

h (HIDDEN LINES): system asks whether or not to hide hidden lines in ensuing displays.

E (ERASE): erase the screen.

B (BINOCULAR IMAGE): display a stereo pair of the polyhedron.

x,y,z,X,Y,Z (ROTATIONS): rotate the polyhedron by small (10 degree) or large (40 degree) amount about screen axes.

r (RESET): reset all magnitudes to SIZE.

m (MAGNIFY): system asks for integer; this becomes SIZE and all magnitudes are linearly scaled so that the largest is SIZE.

d (DISTORT): system asks for a data file and uses it to create a histogram by changing vertex magnitudes. The histogram is scaled so the maximum bin is SIZE and the minimum is BIASDIAMETER. This distorts the shape of the faceted polyhedron.

w (WRITE OUT): plots the contents of the screen on a printer-plotter, magnified by some factor between 1 and 6 (for which the system asks). Many figures of this paper were produced this way.

c (CENTER): system asks for integers giving x and y (screen coordinates) of center of polyhedron.

t (TOP): goes to TOP for brand new faceted polyhedron.

q (QUIT): quits the system.

4. The Techniques.

> So I saw that mathematics were things that could be learned and in order to do a good geodesic you had to do mathematics.
>
> --Buckminster Fuller

This Section should provide enough information that the geometric construction of a few useful varieties of tesselated spheres is made easy. The strategy is to start with an icosahedron, to subdivide each of its faces (actually subdivide a paradigm "principal polyhedral face" and lay it over the icosahedral face), and "punch out" the resulting vertices to the spherical radius. The following geometric information should be of use (and the implementational and hidden-line information may also be of interest) to anyone who wishes to use these methods.

4.1. The Icosahedron.

> In Boticelli's Venus...the line containing the figure from the top of the head to the soles of the feet is divided, at the navel, into the exact proportions given by that ancient formula the 'Golden Section.'
>
> --T.A. Cook

The constant t appearing in the expressions below is the Golden Section, classically defined by

$$\frac{1}{t} = \frac{t}{t+1} .$$

We also note:

$$t = (\mathrm{sqrt}(5) + 1) / 2$$

$$1/t = t - 1$$

$$t^{**}2 = t + 1$$

$$t^{**}4 = 3t + 2$$

$$\mathrm{sqrt}(5)^{*}t = t + 2$$

$$t = \lim_{n \to \infty} (F(n+1)/F(n)),$$ F(n) the nth Fibonacci number

For the icosahedron,

Number of vertices = 12

Number of faces = 20

Number of edges = 30.

Let

Center of icosahedron in Cartesian coordinates = (0, 0, 0)

and

Distance from center to each vertex = 1.

Define the constants

a = sqrt(t) / (5**(1/4))

b = 1/(sqrt(t)*(5**(1/4)))

c = a + 2*b = 1/b

d = a + b = (t**(3/2))/(5**(1/4))

Then:

Angle subtended by edge at origin = A = arccos(sqrt(5)/5)

Angle between radius and an edge = B = arccos(b)

Edge length = e = 2*b

Distance from origin to center of edge = a

Distance from origin to center of face = t*a/sqrt(3)

Area of face = 1 / (t*sqrt(3/5))

Volume = 4/(3*b)

The 12 vertices may be placed at:

(0, +-a, +-b)
(+-b, 0, +-a)
(+-a, +-b, 0);

then midpoints of the 20 faces are given by:

```
(+-d, +-d, +-d)
(  0, +-a, +-c)
(+-c,   0, +-a)
(+-a, +-c,   0).
```

4.2. The Principal Polyhedral Triangle.

The principal polyhedral triangle (ppt) is an equilateral triangle considered to lie in the z=0 plane with its centroid at the origin. It is considered a paradigm of the 20 icosahedral faces when considering subdivision of the icosahedral faces into facets.

Its vertices are at:

```
(+-b,  -b/(sqrt(3)))
(  0, 2*b/(sqrt(3))).
```

In constructing the full faceted polyhedron, we first subdivide the ppt into facets, then rotate and translate the ppt to cover an icosahedral face, and then "punch out" the vertices to be at a uniform radius from the center of the polyhedron.

4.2.1. Coordinatization of the ppt.

If each side of the ppt is divided into N-1 lengths, N = 2,3,4..., then there will be N facet vertices along each side, a total of N*(N+1)/2 vertices and (N-1)**2 facets in the ppt. It is often convenient to use a redundant coordinate system for these facet vertices (Figure 4.1).

INSERT FIGURE 4.1. HERE

Clearly a (facet) vertex is specified by any two coordinates, and in fact for any vertex (i,j,k), i+j+k = N+2. Usually i and j are used to identify points; in this case, to account for all the vertices i must run from 1 to N and j must run from 1 to N-i+1.

4.2.2. Computing Facets I.

Clinton [ClintonNASA] gives seven methods for producing facets from one or two ppts. The simplest of these is to divide the ppt into congruent equilateral triangles. That is, each edge of the ppt is equally divided into N-1 lengths, and vertices occur at the intersections of lines parallel to the ppt sides through these division points (Figure 4.2).

INSERT FIGURE 4.2. HERE

This method has the advantage of simplicity, but the disadvantage that when the ppt is "punched out" onto the sphere (all the vertices are placed at unit distance from the origin), the central facets are pushed out farther than those near the corners of the ppt, and thus give rise to larger triangles. This problem becomes increasingly acute with finer subdivision of the ppt.

The larger facets will clearly subtend more solid angle then the smaller, and the extent of the problem can be measured by comparing the areas of the spherical triangles projected on the unit sphere from the center through the facets. Using figures from [ClintonDome], one can see that for N=5, the ratio of the areas of the largest to the smallest spherical triangle resulting from Method I is about 1.5166.

This problem is the motivation for the second method of subdivision, discussed below (also due to Clinton). With Method II the ratio of areas of largest to smallest spherical triangle drops to about 1.1460 for N=5. Figure 4.3 shows two orthographic projections of punched-out faceted faces for N = 8. One can see that method II produces more uniformly sized facets.

INSERT FIGURE 4.3. HERE

4.2.3. Computing Facets II.

Here the subdivisions of the ppt sides are made to subtend equal angles at the center of the polyhedron. The angle will be (A/(N-1)). The computation of the subdivisions may be done easily by computing the distance from one corner of the ppt to the N-2 vertices along the edge (computing the ln in Figure 4.4).

INSERT FIGURE 4.4. HERE

If C is the angle

$$C = n*A/(N-1),\ n = 1,2,3,...$$

then

$$l(n) = \sin(C)/\sin(C+B)$$

by the law of sines. Clearly $l(N-n-1) = e - l(n)$.

After the vertices along the edges have been located, they are connected by lines parallel to the edges of the ppt. From the two-point formula for a line, the linear form is easily obtained, and the intersection of two lines $a1*x + b1*y = c1$, $a2*x + b2*y = c2$ is given by

$$x = \frac{c1*b2-c2*b1}{a1*b2-a2*b1},$$

$$y = \frac{a1*c2-a2*c1}{a1*b2-a2*b1}.$$

Because the edges of the ppt are not in general evenly divided in method II, the three lines of equal i,j, and k for vertex (i,j,k) do not intersect in a point, but in a small triangle. The average of the three intersection points is taken as the (x,y) point of the vertex.

4.3. The Faceted Polyhedron.

Let n be the number of subdivisions of the ppt edge. Then n = N-1, N the number of vertices along the edge. For the faceted polyhedron,

Number of vertices = 10*(n**2) + 2

Number of faces = 20*(n**2)

Number of edges = 30*(n**2).

4.4. The Representation.

The tesselated sphere, which provides the histogram bins for our microstructural representations, is represented by the faceted polyhedron, in which one imagines great circles connecting the vertices instead of straight lines. With this understanding, we refer henceforth to the representation of the polyhedron.

Internally the faceted polyhedron is represented largely geometrically, with topological information often coded into procedures. For example, one cannot tell a vertex's neighbors by looking at the vertex; one must call a procedure to compute them. The problem of elegantly generating a machine representation which is topologically equivalent to the faceted polyhedron is an interesting one we did not solve.

The polyhedron is initially a list of vertex quadruples of the form [x,y,z,magnitude], which is to be understood as the (x,y,z) coordinates of the vertex (hence the direction cosines of the direction associated with that vertex) and the magnitude by which the (x,y,z) vector is to be multiplied to stretch the point away from the origin (the magnitude is simply the contents of the histogram bin).

Each vertex originally arose from the subdivided ppt which was translated to some particular face F of the icosahedron. This fact allows us to identify a vertex by its "polyhedron coordinates" (F,i,j,k), where F refers to one of the 20 faces, and only two of i,j, and k are needed (Section 4.2.1.).

The faces themselves have associated "spines," which are vectors normal to the planes of the faces; they point from the center of the polyhedron to the centroid of the face.

The (F,i,j,k) scheme of vertex naming means that vertices from the corners of the ppt are represented five times and those from the rest of the edges of the ppt twice. It is seen in fact that if n is the number of subdivisions of the ppt edge, there are 10*(n**2) + 30*n + 20 different names for the 10*(n**2) + 2 vertices. In the system, these multiple representations are collapsed, so that there is only one magnitude for any (x,y,z) vertex. Any vertex has 2, 4, or 6 neighbors within its face. They are computed in terms of their (i,j,k) coordinates for a given (i,j,k) and returned in counterclockwise order (for the hidden line eliminator.)

Thus the polyhedron is geometrically tied together by the unique (x,y,z) representation of all its vertices. However, many topological questions about it cannot be answered easily (inter-face neighbors of a vertex, for instance). The representation is compact, minimizing the topology-preserving pointers, but has had to be extended with other entities, since a polyhedron is more than points.In particular, the full hidden line algorithm requires a list of facets (three pointers to vertices), and a list of lines (two pointers to vertices).

The display routines allow various transformations of the original points to be shown. The perspective transform distorts the geometry in a nonlinear way. Since the hidden line eliminator needs to deal with these transformed points rather than the original structure, and since we typically would like to display the original several times, we have to preserve the original somehow. We could: 1) save the original on disk, distort it in core and retrieve it from disk when needed; 2) make a copy of the original in core and distort the copy; 3) distort the original and invert the distortion to restore the original. We are not committed to any one of these solutions; 1) saves on core, but is unaesthetic, 2) is presently what is done, except that the copy is of the form [X,Y,Z,flags], where (X,Y,Z) = magnitude*(x,y,z) from the original and flags are used by the hidden line eliminator. The best solution might be to use 3) with an extended vertex representation, say [x(X),y(Y),z(Z),magnitude, flags].

4.5. Hidden Lines.

The highly nonconvex polyhedra that can result from the histogramming process are confusing when viewed as wire frame models (no hidden lines eliminated). Such images are ambiguous to within a Necker reversal, which for orthographic projection amounts to a mirror image reversal across the viewing plane. Applying the perspective transform to the wire frames distorts their metric properties; for some simple images this is enough to clamp the figure into the right Necker version, but for complicated unknown objects it is insufficient; further, it leaves the confusion of crossing lines in the image. The obvious answer is to remove hidden lines.

This well-known problem was made a bit difficult for us by our representation of the faceted polyhedron, by the large number of facets we had to deal with, and by the limited core capacity of our machine. Four versions of the problem were solved, and below we outline the techniques and our opinion of them.

4.5.1. Elimination of Invisible Faces.

Organize the display of facets by faces, and display only those facets whose face is facing the observer (whose spine has a component in the direction of the observer). This method is only completely correct for icosahedra. In faceted polyhedra, the icosahedral faces have been punched out onto the sphere, and have been "bent around the horizon"; thus even for convex faceted polyhedra this method fails by omission and commission. It makes no pretence of handling non-convex polyhedra, but does eliminate many lines arising from the back of the object. It is incredibly cheap to do, and introduces no new data structure.

4.5.2. Elimination of Invisible Facets.

Display all and only facets which are facing the observer (called the potentially visible facets). This method is completely correct for all convex faceted polyhedra. It requires some work to extract facets from the data structure and assign normals to them, but requires only minimal new storage.

4.5.3. Elimination of Partially Hidden Lines.

Compute the potentially visible facets and save a list of them. For every vertex implicit in this list, check to see if it is behind a facet in the list. If so, eliminate all lines originating at it. Performing the check is speeded up by first checking to see if the point is within the rectangle surrounding the triangular 2-D image of the facet (a few fixed point compares are needed). If a point is within rectangle and within the triangular image of the facet, 3-D routines determine if it is behind the facet. If it is, no lines emanating from it are drawn.

This method takes a fair amount of geometric computation, and needs a list of potentially visible facets; it does not worry about any lines which are not between facet vertices, so it does not construct any new line endpoints.

The method is completely correct in no larger a class of problems than the method of 4.5.2. It fails by omission and commission, as illustrated by the examples of Figure 4.5.

INSERT FIGURE 4.5. HERE

It was vainly hoped that these failures would be infrequent or at least non-irritating. In the event, we could not ignore them, and were forced to confront the Hidden Line Problem in its full generality.

4.5.4. Elimination of Hidden Lines.

There is an immense literature on the hidden line problem. We considered purely geometric methods in which every line of the potentially visible facets is compared with every facet in 3-D. If a line is partially hidden, the hidden parts are removed. The raster display allows easy erasure of parts of lines, so there is no need to create and remember for very long a list of new vertices at the endpoints of visible line segments.

The calculation involved in the geometric scheme is prodigious, so we went on to implement a method that takes cognizance of the discrete 2-D display device as well as the continuous 3-D nature of the data. This is the Warnock algorithm, which concentrates its attention on "interesting" parts of the image rather than comparing everything against everything else. This algorithm produced Figure 2.8-B. More on these issues can be found in [Newman and Sproull].

5. The Summary.

In this perhaps distressingly wide-ranging paper we have started with a particular neurophysiological application of statistics of 3-D vector data. Such data also arises in meterology, geology, oceanography, cardiology, and elsewhere. Two attempts to describe the orientation properties of the data were presented, one (macrostructural) a gross characterization of spatial extent, the other (microstructural) simply a 3-D histogram of the vector data.

Since there is some interest in configuring computer systems for neurological research, the two systems used in this study are described.

Since the basis of the 3-D histogram, the geodesic construction, is commonly seen, some of the details of its structure are presented in the spirit of a technical appendix.

Acknowledgements.

Thanks to P.D. Coleman for providing the resources of the dendrite-tracking and analysis system, J. Wellner for statistical guidance, D. Bliss for coping with ALGOL, L. Scott for implementing method II, C. Meltzer and K. Lantz for implementing various hidden line eliminators, and P. Hall for caring. This research was sponsored by Alfred P. Sloan Foundation Grant No. 74-12-5.

References.

Brown, C. Principal axes and best-fit planes, with applications. Milwaukee Symposium on Automatic Computation and Control, Milwaukee, 1976.

Christie, D.E. Vector Mechanics. McGraw-Hill, New York, 1964.

Clinton, J.D. Advanced structural geometry studies part I: polyhedral subdivision concepts for structural applications. NASA CR-1734/35, (Sept. 1971).

Clinton, J.D. Geodesic math. in Domebook 2. Pacific Domes, Calif, 1971.

Creutzfeldt, O.D. and Heggelund, P. Neural plasticity in visual cortex of adult cats after exposure to visual patterns. Science, 188 (June 1975).

Fuller, R.B. Building Construction, patent no. 2,682,235, (1951).

Garvey, C.F., Young, J.H., Coleman, P.D., and Simon, W. Automated three-dimensional dendrite tracking system. Electroenceph. and Clinical Neurophysiology, 35 (1973).

Hubel, D.H. and Wiesel, T.N. Receptive fields, binocular interaction and functional architecture in two non-striate visual areas (18 and 19) of the cat. Journal of Physiology, 160, (1962).

Mardia, K.V. Statistics of Directional Data. Academic Press, New York, 1972.

Newman, W.M. and Sproull, R.F. Principles of Interactive Computer Graphics. McGraw-Hill, New York, 1973.

Pearson, K. On lines and planes of closest fit to systems of points in space, Philosophical Magazine, 6th series,2, July-Dec., 1901.

Reddy, D.R., Davis, W.J., Ohlander, R.B., Bihary, D.J. Computer analysis of neuronal structure. In Intracellular Staining in Neurobiology, Kater and Nicholson, Eds., Springer-Verlag, New York, 1973.

Richards, M. BCPL: A tool for compiler writng and systems programming. Spring Joint Computer Conference, 1969.

Siegel, S. Nonparametric Statistics. McGraw-Hill, New York, 1956.

Wann, D.F., Woolsey, T.A., Dierker, M.L., Cowan, W.M. An on-line digital-computer system for the semiautomatic analysis of Golgi-impregnated neurons. IEEE Transactions on Biomedical Engineering; BME-20, 4, (July 1973).

Figures.

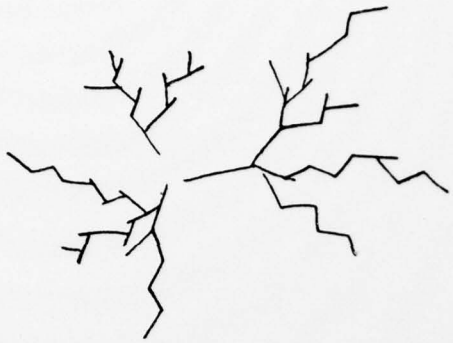


Figure 1.1. A rod tree (to be imagined in 3-D).

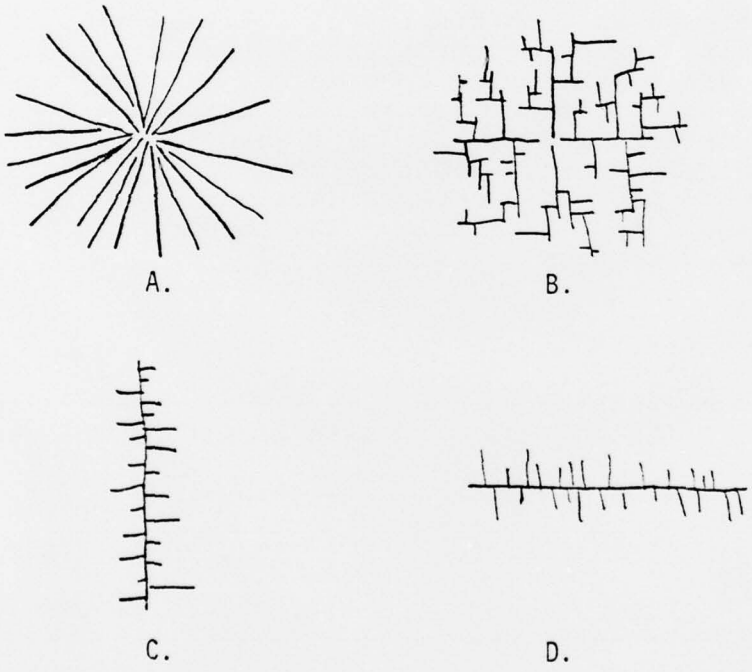


Figure 1.2. Macrostructure and microstructure. A and B have the same macrostructure ("pancake") but different microstructure ("many directions" vs. "four directions"). C and D have the same microstructure ("four directions") but different macrostructure ("vertical cigar" vs. "horizontal cigar").

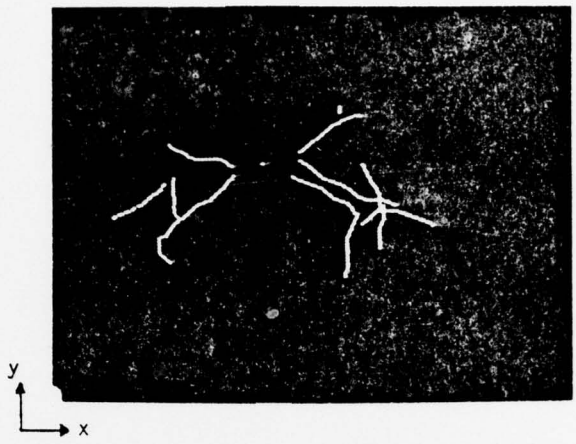


Figure 2.1. Principal component analysis.

Photograph is from display of rod tree data from automatic dendrite tracking program. The z axis is out of the page. Rods are approximated here as weighted points at C. M. of rods. Is, Icm: (relative) principal moments about axes through soma and C. M. of system. Ps, Pcm: (relative) principal moments about planes through soma and C.M. Here axis of minimum inertia (maximum dispersion) is nearly parallel to the x axis. The dispersion is less pronounced for axes through the soma.

Mass = 464.78

Axis	Is	Ps	axis direction cosines
1	1.00	1.79	(-.965, .043, .255)
2	2.39	.40	(.234, .559, .795)
3	2.19	.60	(.108,-.828, .550)

Axis	Icm	Pcm	axis direction cosines
1	1.00	2.11	(.951, .053,-.303)
2	2.58	.53	(.307,-.234, .992)
3	2.64	.47	(.022, .970, .239)

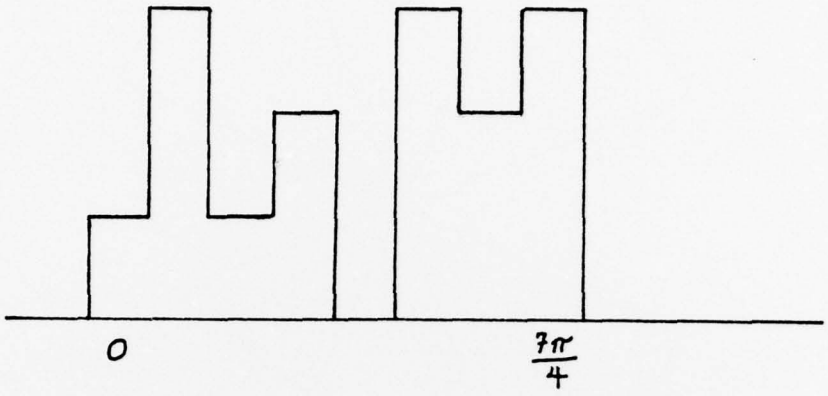


Figure 2.2. A histogram of 2-D directions.

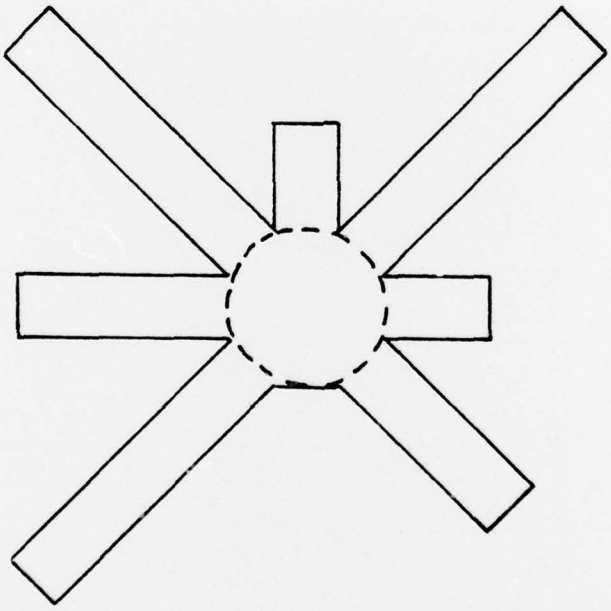


Figure 2.3. A circular histogram of the data of Figure 2.2.

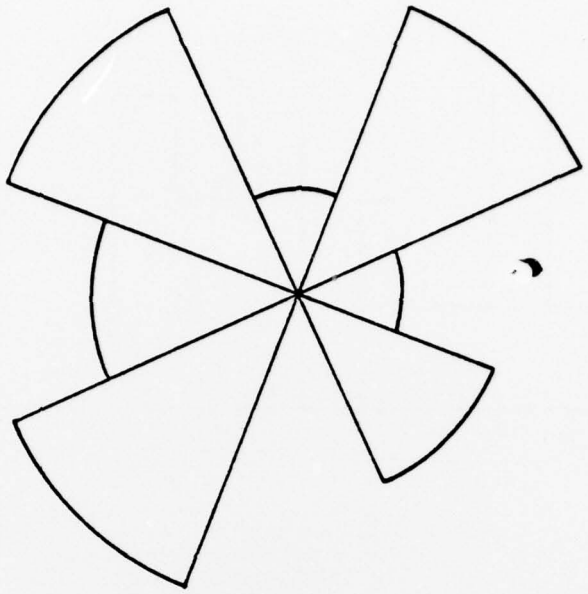


Figure 2.4. A rose diagram of the data of Figure 2.2.

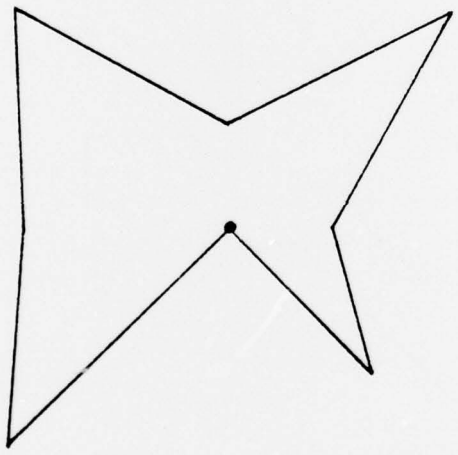


Figure 2.5. Histogram of data of Figure 2.2. analogous to 3-D histograms (Figure 2.8.).

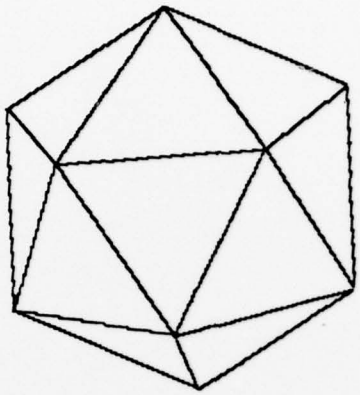


Figure 2.6. The icosahedron.

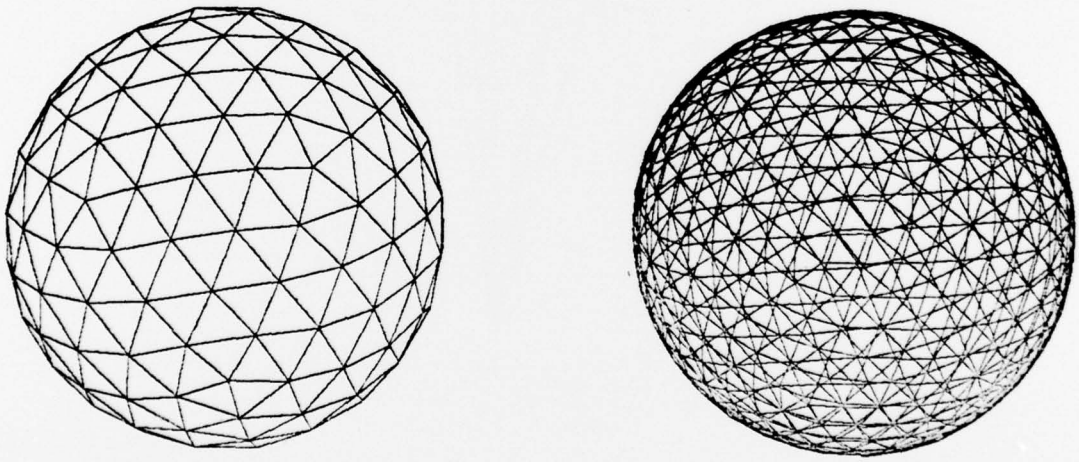


Figure 2.7. Faceted polyhedra, shown as solid and wire frame objects; the icosahedral faces have been divided into 16 and 49 facets, respectively. They provide useful tesselations of the sphere.

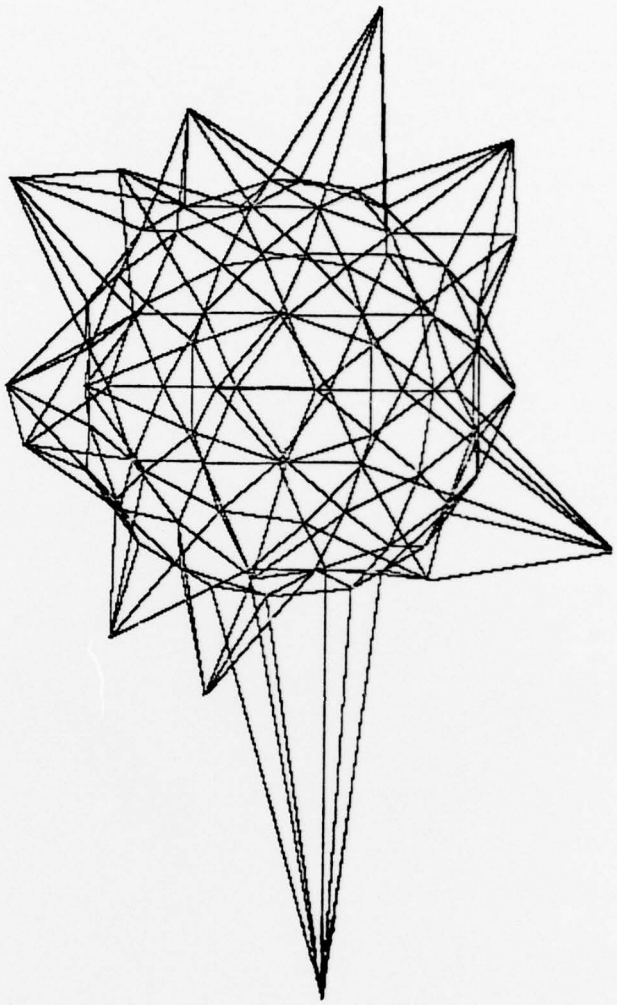
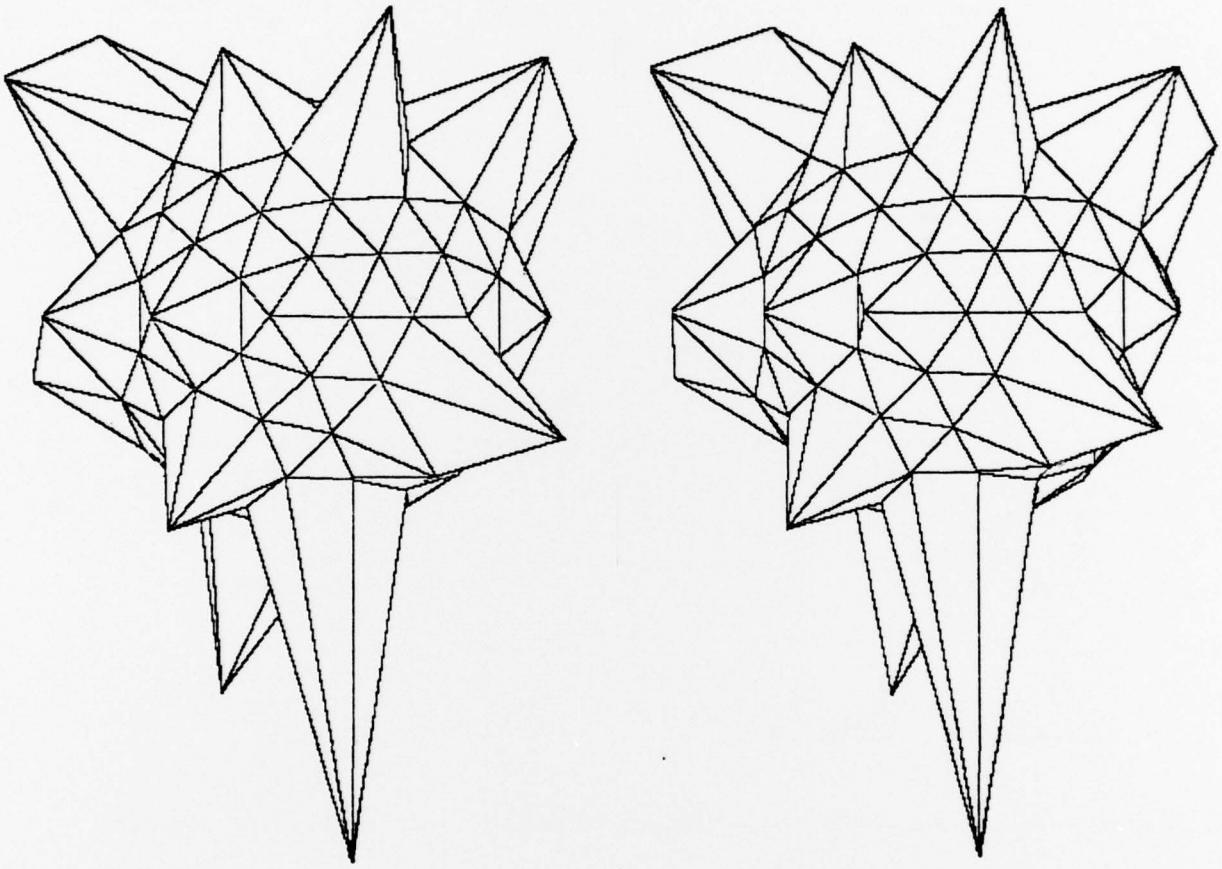


Figure 2.8. Distorted faceted polyhedra. A is an orthographic projection of a wire frame, B (see next page) a stereo pair using perspective distortion and a version of a hidden-line algorithm.



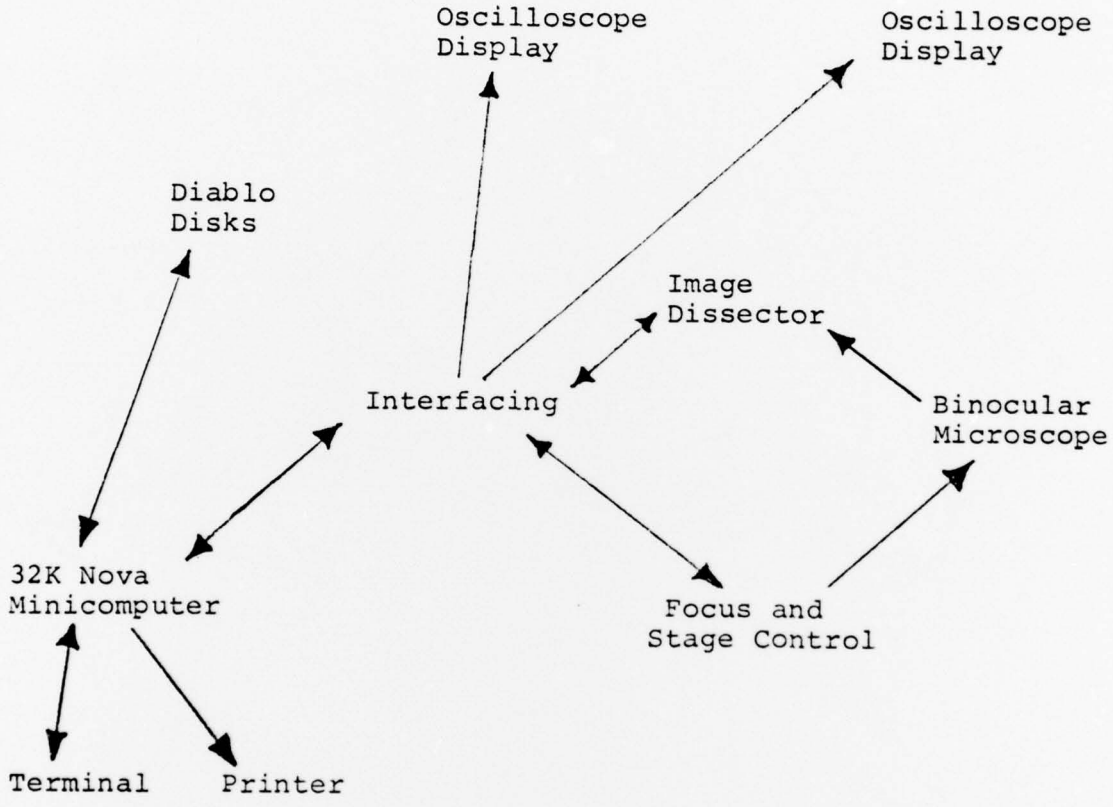


Figure 3.1. Automated dendrite tracking and analysis system.

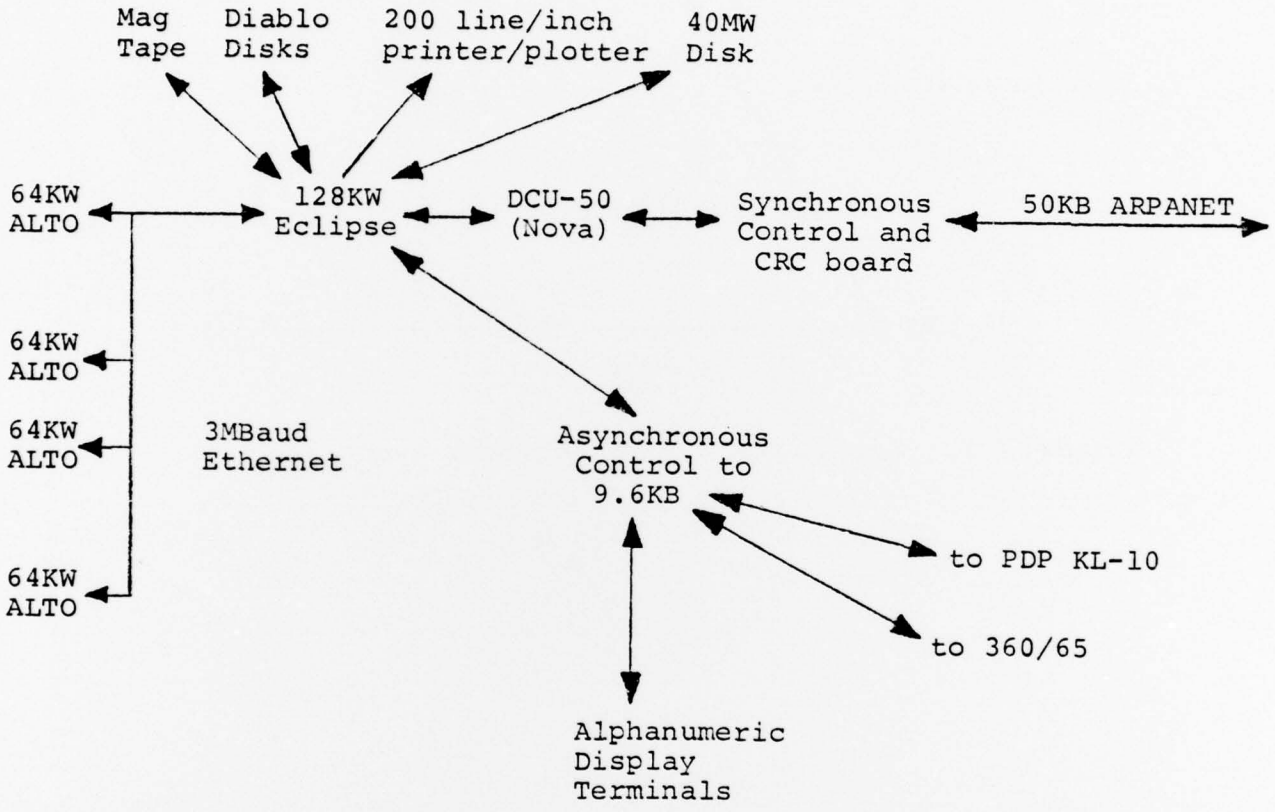


Figure 3.2. Computer Science Laboratory hardware facilities.

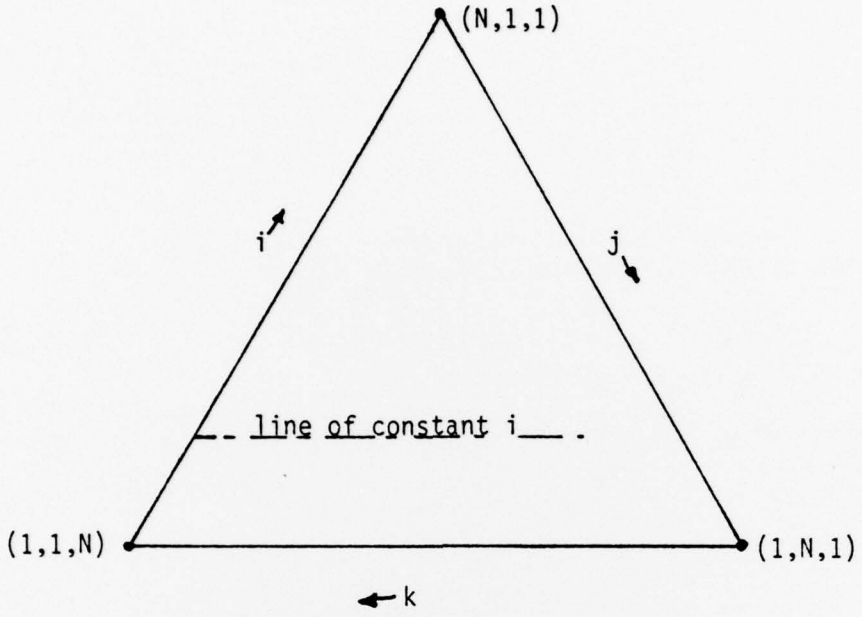


Figure 4.1. Coordinate system for facet vertices.

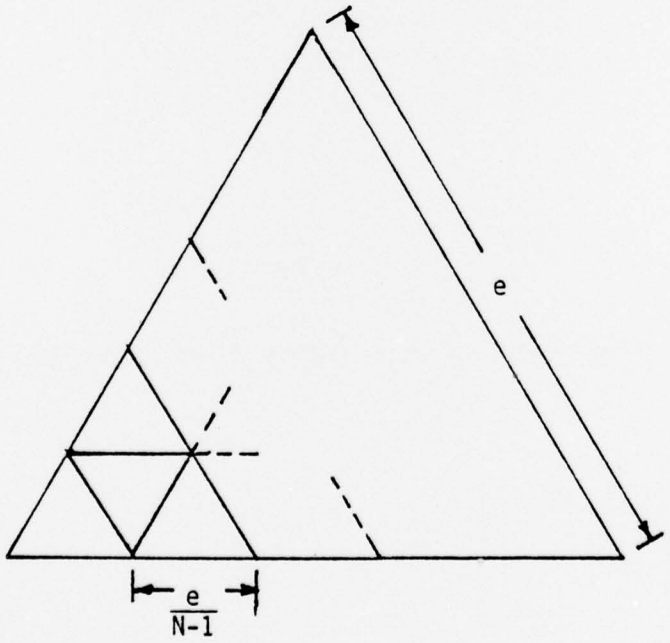


Figure 4.2. Subdivision method I; congruent equilateral triangles.

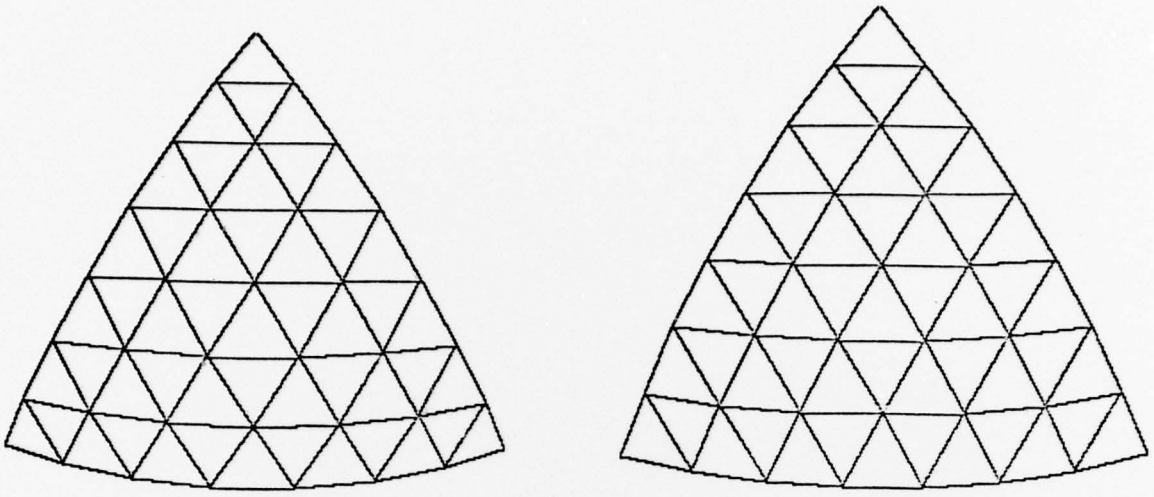


Figure 4.3. Orthographic projection of facets on sphere. Method I on left, Method II on right, N = 8.

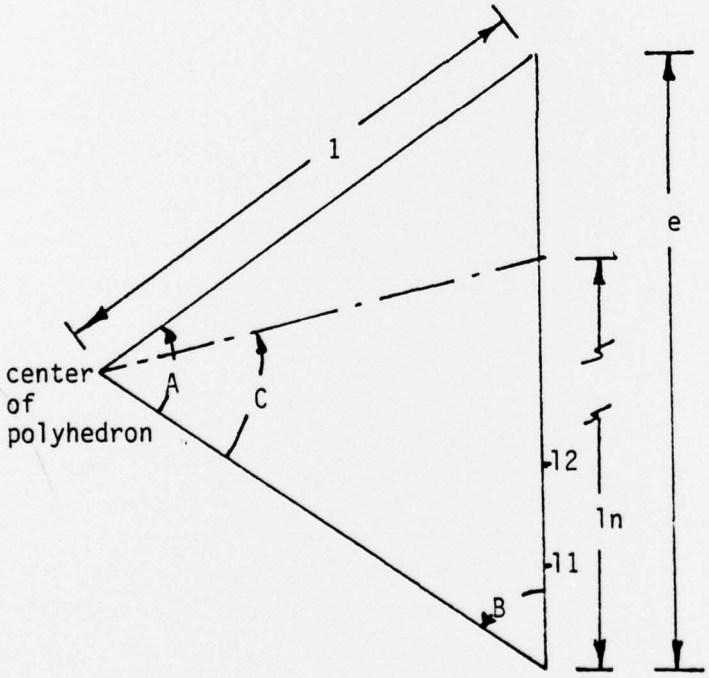


Figure 4.4. Computation of edge subdivisions, Method II.

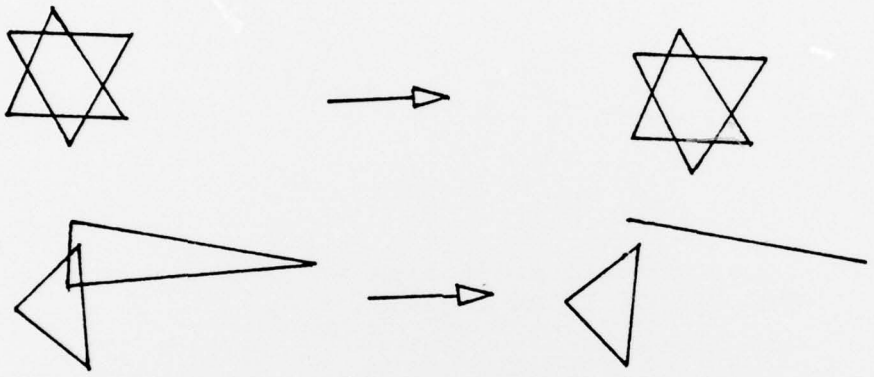


Figure 4.5. Failures of a hidden-line eliminator.